Cross Validation

Cross-validation is a model validation technique for assessing how the results of a statistical analysis will generalize to an independent data set. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice. In a prediction problem, a model is usually given a dataset of known data on which training is run (training data set), and a data set of unknown data against which the model is tested (testing data set). The goal of cross-validation is to define a dataset to "test" the model in the training phase (i.e., the validation data set), in order to limit problems like overfitting, give an insight on how the model will generalize to an independent dataset (i.e., an unknown dataset, etc.

Setting

- $(X, Y) \sim \wp$
- Dataset:

$$\begin{pmatrix} X_1 & Y_1 \\ . & . \\ . & . \\ . & . \\ X_N & Y_N \end{pmatrix}$$

- M(X) prediction model (trained with our dataset)
- Validation summary $Z(M, \wp)$
- Estimate of $Z(M, \wp)$: $Z(M, \widehat{P})$, where $\widehat{P}$ is the empirical distribution
- Drawback: in most cases $E[Z(M, \wp)] < E[Z(M, \widehat{P})]$

Implementation

1. Training set:

$$\begin{pmatrix} X_1 & Y_1 \\ . & . \\ . & . \\ X_{i-1} & Y_{i-1} \\ X_{i+1} & Y_{i+1} \\ . & . \\ . & . \\ X_N & X_N \end{pmatrix}$$

2. Testing set (X_i, Y_i)
3. Estimate of Z: $Z(M^{(-i)}, \widehat{P}_i)$ where: $M^{(-i))}$ denotes the model trained on all the N data points except (X_i, Y_i) and $\widehat{P}_i$ denotes the validation point i.

- Repeat steps 1, 2, and 3 for each i, with $i = 1, ..., N$.
- Cross Validation estimate:

$$\frac{1}{N} \sum_{i=1}^{N} Z(M^{(-i)}, \widehat{P}_i)$$

Example

Let's suppose that we want to get an estimate of R^2 using a cross validation method. We will use the `stackloss` data set. This built-in R dataset has measurements on 21 days of operation of a plant for the oxidation of ammonia (NH3) to nitric acid (HNO3).

`Air.Flow` represents airflow to the plant

`Water.Temp` is the cooling water inlet temperature

`Acid.Conc.` is the acid concentration as a percent (coded by subtracting 50 and then multiplying by 10)

`stack.loss` is the percent of ammonia lost (times 10)

More information can be seen at https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/stackloss.html

```
library(cvTools)
data = stackloss
head(data)
```

```
##   Air.Flow Water.Temp Acid.Conc. stack.loss
## 1       80         27         89         42
## 2       80         27         88         37
## 3       75         25         90         37
## 4       62         24         87         28
## 5       62         22         87         18
## 6       62         23         87         18
```

First, decide how many folds you wish to use, and call that number `k`. Then use the `cvFolds` function to split the data into the specified number of folds.

```
attach(data)
k = nrow(data)                          # The number of folds we want
folds = cvFolds(NROW(data), K=k)        # This function splits the data into folds
data$holdoutpred = rep(0,nrow(data))
```

Now, loop through each data set (fold), designate training and validation sets, fit the desired linear regression model, and record the predictions.

```
for(i in 1:k){
train = data[folds$subsets[folds$which != i], ]       # This is the training set
validation = data[folds$subsets[folds$which == i], ] # This is the validation set
mod =lm(stack.loss ~ Air.Flow + Water.Temp + Acid.Conc., data=train)
pred = predict(mod, newdata = validation)
data[folds$subsets[folds$which == i], ]$holdoutpred = pred
}
```

Now, calculate the R^2 estimate value after cross validation has been performed.

```
dif = abs(data$stack.loss - data$holdoutpred)
r.sq = 1 - mean(dif^2)/var(data$stack.loss)
r.sq
```

```
## [1] 0.8656653
```